

AD-A051 396

TRACOR INC AUSTIN TEX  
SONARCON SYSTEM FOR THE UNIVAC 1108. (U)  
JAN 67

F/G 17/1

UNCLASSIFIED

TRACOR-67-137-U

NL

1 OF 1  
AD  
A051 396



END  
DATE  
FILMED  
4-78  
DDC

*Good m-*  
TRACOR Document Number  
67-137-U

— MOST Project -4

0

**OCW LIBRARY COPY**

002654

AD A051396

SONARCON SYSTEM FOR THE UNIVAC 1108

Preliminary Report

January 18, 1967

AD No. 1  
AD FILE COPY  
DDC FILE COPY

D D C  
REPORTED  
RECEIVED  
MARCH 16 1978  
FUGITIVE  
D  
*[Signature]*

6500 Tracor Lane, Austin, Texas 78721, AC 512/926-2800

**TRACOR**

*1/2*  
**DISTRIBUTION STATEMENT A**  
Approved for public release;  
Distribution Unlimited

**TRACOR**

6500 TRACOR LANE, AUSTIN, TEXAS 78721

(14)

TRACOR Document Number  
67-137-U

(2)

SONARCON SYSTEM FOR THE UNIVAC 1108.

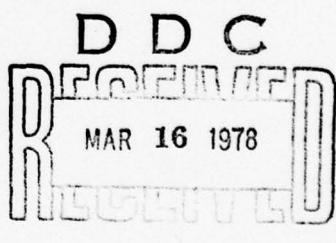
(9) Preliminary Report

(11)

18 January 18, 1967

(12) 27 p.

ABSTRACT FOR	
DTG	White Section <input checked="" type="checkbox"/>
DDG	Surf Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
IDENTIFICATION	Per Mr. on file
BT	
DISTRIBUTION/AVAILABILITY CODES	
FILE	AVAIL and/or SPECIAL
P	



Programmers:  
H. R. Courts  
J. G. Fairey  
J. C. Humphrey

**DISTRIBUTION STATEMENT A**

Approved for public release;  
Distribution Unlimited

352-190

SP

TABLE OF CONTENTS

	<u>PAGE</u>
Introduction	1
General description of the system	2
Processing of run by SONARX (Flow chart)	5
Processing within section 3 (Flow chart)	6
Data storage and handling	8
Black box coding conventions	10
Instructions for use	15
Input deck	15
Topology cards format	16
Data cards format	17
Error diagnostics	19



6500 TRACOR LANE, AUSTIN, TEXAS 78721

## INTRODUCTION

SONARCON is a control program, ~~written at TRACOR, in Austin,~~ for use on the UNIVAC 1108 computer, to facilitate the processing of sonar data.

The program was originally designed and coded for the CDC-3200 computer. The present system is a thorough revision and considerable extension of that system.

Included herein is a preliminary discussion of the formulation, application, and usage of the new SONARCON.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

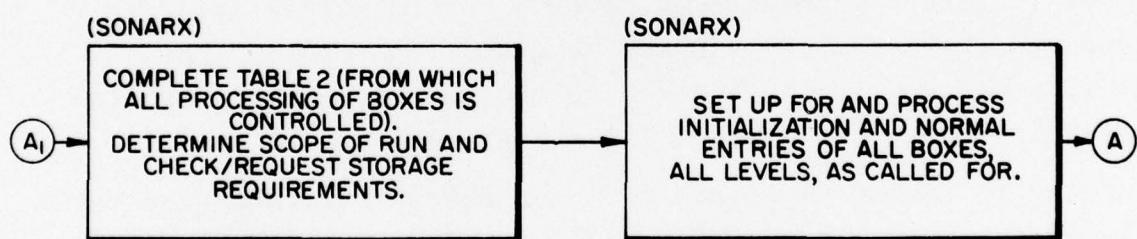
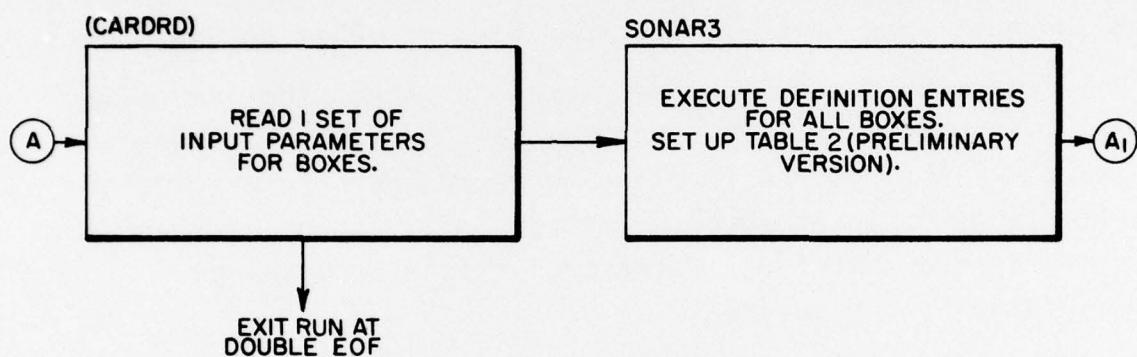
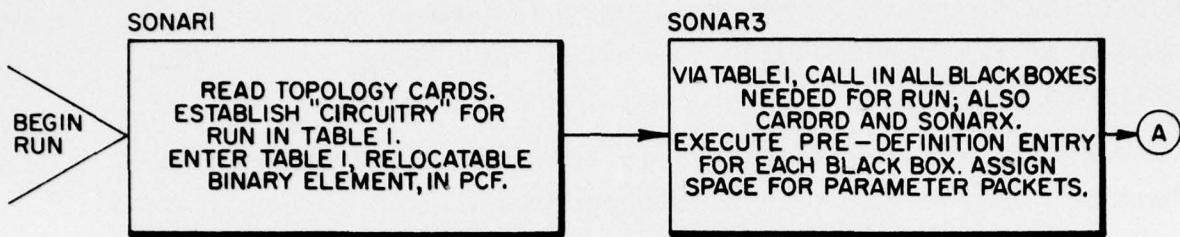
## GENERAL DESCRIPTION OF THE SYSTEM

The new system is composed of three subsystems, SONAR1, SONAR2, and SONAR3 as indicated in the diagram following (p.3). SONAR1 reads the topology list and prepares a relocatable element in the user PCF. SONAR2 is a processed map used to load SONAR3; its only purpose is to force SONARX to be the last element allocated. (See page 8.) SONAR3 includes CARDRD, the sub-routine which reads in and checks all the box input parameter cards, and SONARX, the main processing portion of the system.

The basic structural concepts of a SONARCON problem are unchanged in the new system--levels and sections retain the same meaning; control passes through the levels and sections essentially as before. Any number of data sets may be run through the same topology.

The structure of the actual processing of time-function data through the boxes has been simplified. The number of data samples per record is not fixed; it varies dynamically according to the results of each box processed. One input record per channel produces one output record per channel, although the numbers of samples in an input record and an output record need not be equal, and either may be void. (It is required, however, that a number of samples be available on each input time-function channel to a box equal to that on every other input time-function channel to that box, and that all output time-functions from a given box have equal numbers of samples. See Black Box Coding Conventions.) All samples actually produced are processed forward as a complete record through the section. Hence, the need for "partial-record forking" and "telescoping" has been eliminated.

Time functions are read by a black box, MTI, and/or generated by some box such as SINGER or PIP. The number of samples per record to be delivered to the SONARCON run is specified by the user;



there need be no relationship between this number and the number of samples in the physical record on the tape being read. (See MTI write-up.)

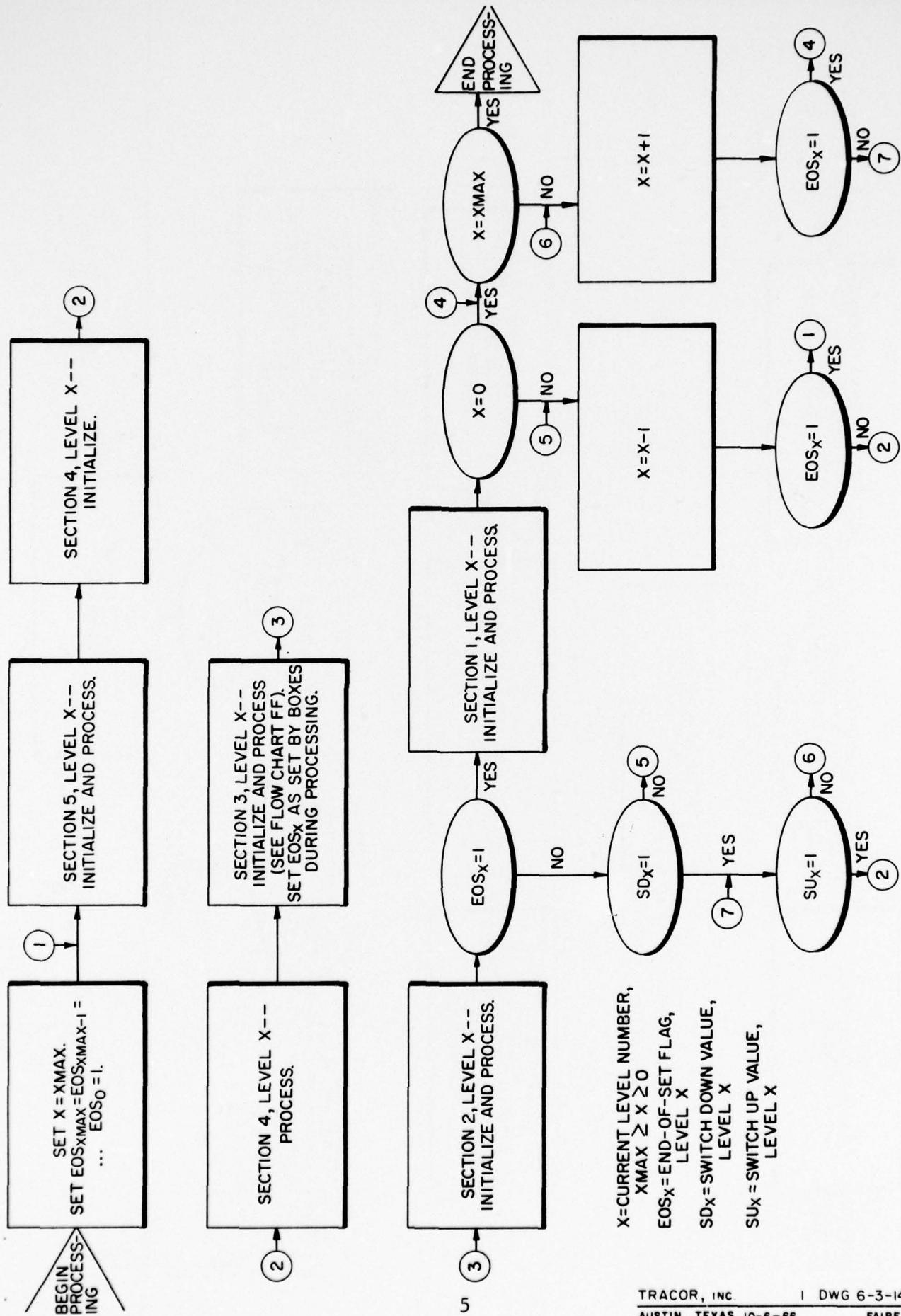
Processing through section 3 is terminated, as before, when the last box in the section has been processed and an end-of-file mark (end of echo) has been sensed and flagged.

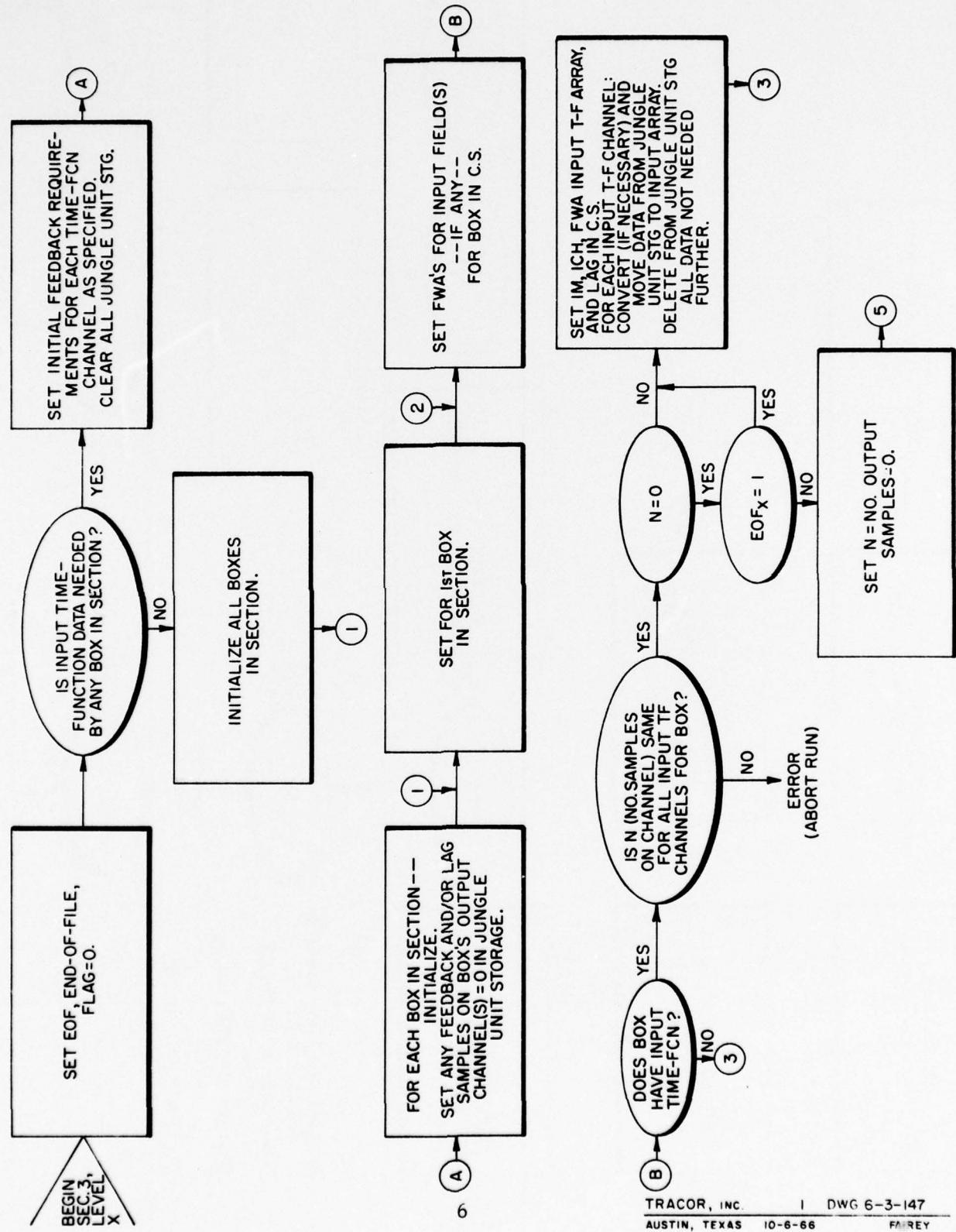
N, the number of files to be processed through the level, is specified by the user to the time-function data input or data generation box. The value is not specified directly to SONARCON.

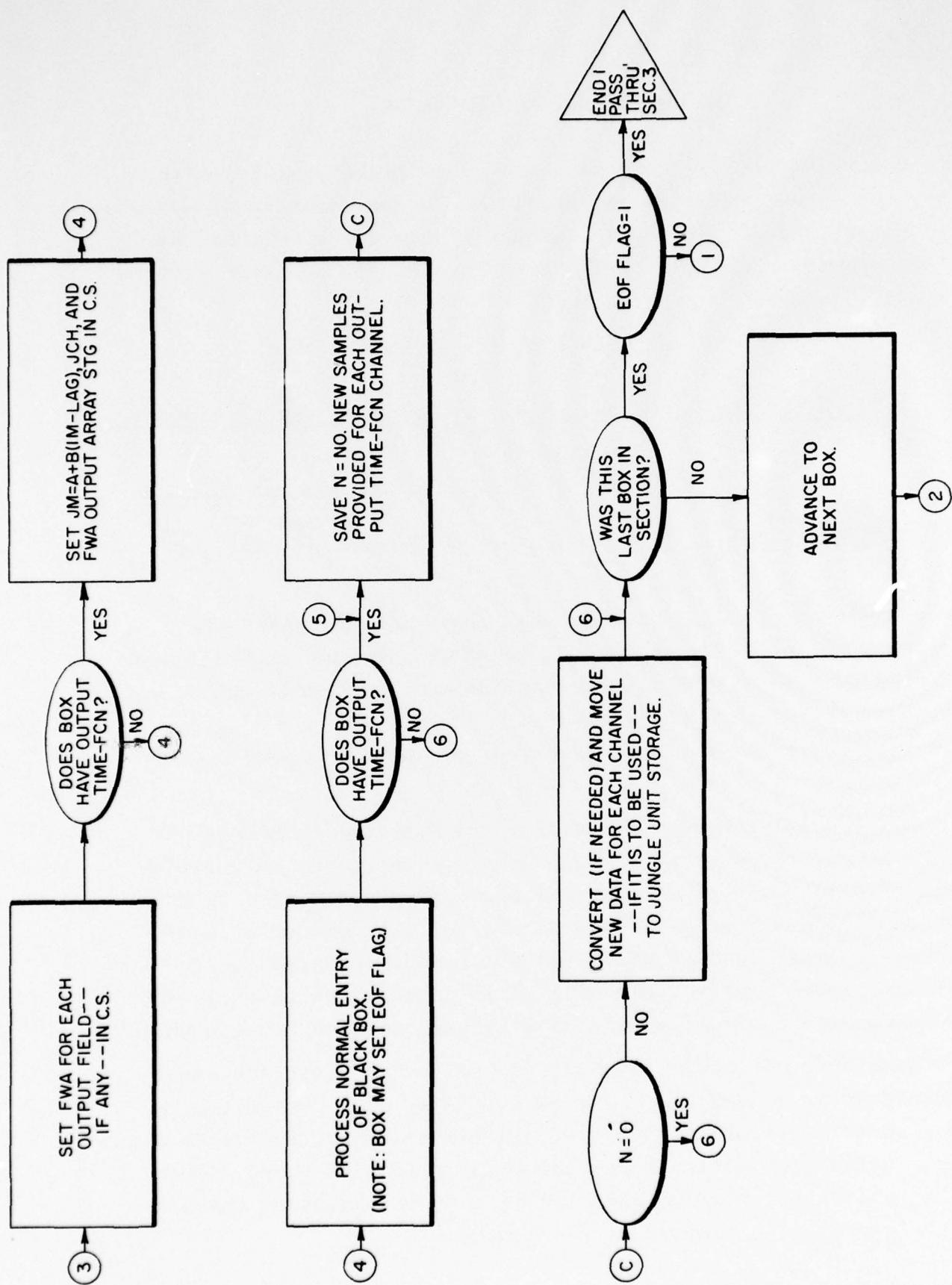
The switch up and switch down settings effect the same flow of control through multi-level problems as before. A setting of 1 causes all files for the level to be processed before going to the next higher or lower level; a setting of 0 causes a transfer of control to the next level after each file's processing. Standard values for both are 1.

A further capability, that of feedback, has been added to the new system. That is, a box may request time-function input data that is generated by itself or by a box occurring later in the topology (but still within the same section three). The feedback requirement must be given in the topology along with the corresponding input channel specification. (See page 17.)

On the following pages are two flow charts indicating in more detail how the actual processing progresses through SONARX.



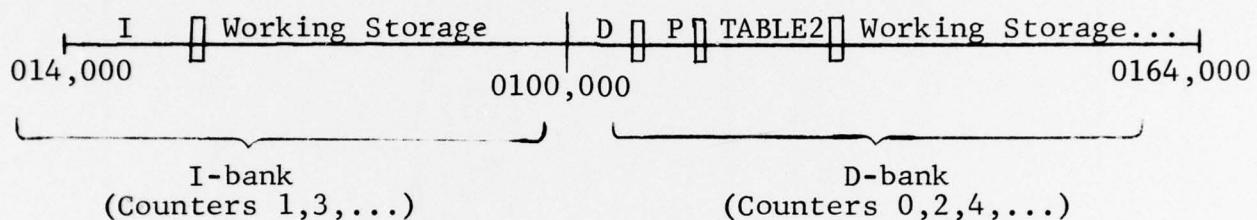




## DATA STORAGE AND HANDLING

Under EXECII, the current resident software system for the 1108, multiprogramming is not possible. Hence, storage requirements need not be specified to the resident system for each run; the entire memory, exclusive of the areas needed for computer systems programs, is available to SONARCON.

The allocation of available memory is as follows:



where I represents all instructions, including the SONARCON programs, all black boxes needed, and all attendant sub-routines; D represents the data areas for the programs; and P is the area assigned for the parameter packets.

Note that the sizes of these areas and of TABLE2 vary with each run.

All remaining memory is assigned at run time by SONARX, in order, to (1) the working storage area for input and output time-function data for a given box (the maximum amount needed by any one box in the topology is determined, and this amount of consecutive storage is assigned), (2) all fields required for the run (all data for a single field will be in consecutive memory), and (3) intermediate time-function data storage, called "jungle units".

High core, following TABLE2, is considered first for each block of storage, and the assignment is made there providing the block under consideration fits within the remaining available area. Jungle units then fill all remaining space in both banks. This scheme allows for minimum run time in that it maximizes the use of the overlapping feature of the computer.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

Field data is all carried in unpacked form, either fixed or floating, as supplied by the box that generates the data. Before processing begins, storage requirements are determined and locations for all blocks of field data are assigned. These are then fixed for the run. Once generated, a field is saved throughout the processing of that entire data set.

Time-function data may appear in SONARX in any of three forms--fixed point, packed (12-bits/sample, 3 samples/computer word), fixed point, unpacked (36-bits/sample), or floating point (36-bits/sample).

Data is read from and written on tape (by MTI, MTO) in packed, fixed point form. It is supplied to, and returned to the system by, each black box in unpacked fixed or floating point form. In intermediate (jungle unit) storage, the data is either in packed fixed point form, or in floating point form. (Floating point form is not allowed in intermediate storage when the data has been generated in fixed point or when all references to the channel call for fixed point data.)

Intermediate storage is arranged in "jungle units." Each unit consists of  $24_{10}$  computer words, including one control word and 23 data words (23 floating point samples, or 69 packed fixed point samples). As many units as needed to hold a complete channel, including any necessary lag, are chained together by means of the control words. Thus, any number of variable length channels can be stored and not have to be moved within the storage area at any time. Old samples can continually be deleted from a channel, and new ones added, but it is never necessary to move data within intermediate storage. All data is deleted from storage as soon as it is used for the last time; only those portions of time-functions needed for further use are saved.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

## BLACK BOX CODING CONVENTIONS

Each black box consists of four sub-routines as indicated below. Let XXXXX (five or fewer alphabetic\* characters) be the base name of the box, exclusive of the use number, then the sub-routines are designated as follows:

XXXXX0	Pre-definition
XXXXX1	Definition
XXXXX2	Initialization
XXXXX3	Normal

Each of these sub-routines is described below. The formats discussed assume FORTRAN black boxes. Machine language, with compatible calling sequences, may be used. The instructions for machine language programs should be assembled under counter 1, and data under counter 0. These counter assignments are automatic under FORTRAN.

All calls are made for each use of each black box referenced.

### Pre-definition

The call to the pre-definition entry is made after reading the topology list and before reading the card input.

The purpose of the sub-routine is to define the size and location of a storage packet to hold parameters needed by the black box. Standard values may be set within the packet as desired.

---

\*The actual restrictions are slightly less. Only the first and last characters of the box name must be alphabetic, or, if the first letter is a "P" or "R", the first two and the last must be alphabetic. The total number of characters in the box name including the use number must not exceed seven. For example, ABCD999, AB6CD2, and ABCDEF1 are all valid box-plus-use-number names; ABCDEF12, P6CD2, and 2XYZ3 are not.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

The format of the pre-definition call as issued by SONAR3 is:

```
CALL XXXXX0(N,P)
```

where N is the location of the number of words required in the parameter packet and P is an array which is the packet. (N) may be set to zero. At the time of the call, an adequate block of storage is allotted to the array. Upon return from the call, SONAR3 collapses the storage block to (N) words.

The format of the black box pre-definition sub-routine is:

```
SUBROUTINE XXXXX0 (N,P)
DIMENSION P(1)
N = (Required value)
(Optional code to set any standard
values in the parameter packet)
RETURN
```

#### Definition

The definition call occurs after card input and therefore can use any of the input parameters it may need.

Format of the call is:

```
CALL XXXXX1(P,INPF, INPT, KINPT,L,IOUTF,I,IOUTT,KOUTT,A,B)
```

where:

P	Parameter packet
INPF	Number of input fields required
INPT	Number of input time functions required
KINPT	Flag specifying fixed (KINPT=0) or floating (KINPT=1) point time function data input. SV=0
(Omit if INPT=0) {	
L	Number of lag samples required on each input time function

IOUTF Number of output fields to be provided  
(Void if IOUTF=0) { I(j), j=1,2,...,IOUTF  
                  Number of data words in j-th output field  
                  IOUTT Number of output time functions  
                  (Void if IOUTT=0) { KOUTT Flag indicating that all output time function  
                                  samples from box are fixed (KOUTT=0) or floating  
                                  point (KOUTT=1). SV=0  
                  A,B Floating point numbers used to determine the  
                                  maximum number of samples of time function  
                                  output, M, provided on a single normal entry.  
                                  SONARX computes -  
                                   $M = A+B \cdot N$   
                                  where N is the number of the input samples  
                                  provided.

Format of the sub-routine is:

```
SUBROUTINE XXXXX1(P,INPF,INPT,KINPT,L,IOUTF,I,IOUTT,KOUTT,A,B)
DIMENSION P(1), I(1)
(Code required to set values for all formal parameters
except P.)
RETURN
```

#### Initialization

The format of the initialization call is:

```
CALL XXXXX2(P)
```

The box may set initial values in the P array as required.  
The sub-routine format is:

```
SUBROUTINE XXXXX2 (P)
DIMENSION P(1)
(Optional code)
RETURN
```



6500 TRACOR LANE, AUSTIN, TEXAS 78721

Normal

The structure of the normal call is dependent on the values provided for INPF, INPT, IOUTF, and IOUTT at definition. While these parameters may vary from one box to another, so long as they remain constant for a given box, it is possible conveniently to structure the normal sub-routine at program time. The general format of the call is:

```
CALL XXXX3(P,K,FINP1,FINP2,...,  
          {INPF values. Omit if INPF=0 }  
  
IM,INPT,AI,L,FOUT1,FOUT2,...,  
          {Omit if INPT=0 } {IOUTF values. Omit if IOUTF=0 }  
  
JM,IOUTT,AJ,N)  
          {Omit if IOUTT=0 }
```

where:

P        The parameter packet  
K        End-of-echo and/or end-of-set flags.  
          Bit 0=1 flags the end of the echo cycle (end  
          of file) bit 1=1 indicates end of set of  
          files (echo set). These flags may be set by  
          any box capable of making such a determination,  
          e.g., MTI, SCEXT, RFEXT.  
FINP1,... The locations of the input fields.  
          The number of these is equal to INPF.  
IM       Total number of samples in each input time  
          function record (including lag)  
INPT      Number of input time functions  
AI        Two-dimensional array (IM,INPT) which provides  
          all time-function input  
L        Number of lag samples



6500 TRACOR LANE, AUSTIN, TEXAS 78721

FOUT1,....The locations of the output fields. The number of these is IOUTF

JM The maximum number of output samples allowed; SONARX evaluates this as  $JM=A+B(IM-L)$

IOUTT Number of output time-function channels

AJ A two-dimensional array (JM,IOUTT) which will receive all time-function output

N Value set by the box to the number of output time function samples actually provided on each channel.

If  $N=0$ , SONARX continues processing, skipping any boxes using this output as input, except on an end-of-file pass, when all boxes are processed.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

## INSTRUCTIONS FOR USE

### Input Deck

The following cards are generally needed for a SONARCON run:

```
7 RUN ABC001,T22022,tt,pp          *
8 ASG A=SCN                         **
7 XOT CUR
8 TRW A
IN A                                 ***
7N XQT SONAR1
Topology Cards:
:
7 EOF
8
7 XQT SONAR2
Data Set 1
7 EOF
:
Last Data Set
7 EOF
7 EOF
```

---

\* See 1108 Center Bulletin #3 for I.D. and accounting fields format.  
tt=maximum running time in minutes (SV=5) and pp=maximum number  
of pages of output (SV=50).

\*\* Any other tapes needed for the run should be assigned similarly,  
and the cards included here.

\*\*\* If the run is to include any reassemblies, a second IN card  
should be included here (calling in the symbolic codes), followed  
by the appropriate ASM or F0R control cards and decks.



6500 TRACOR LANE, AUSTIN, TEXAS 78721

### Topology Cards Format

The topology list consists of one card for each black box (including use number) to be used. The boxes must be listed in an executable order.

The format for each card is as follows:

Column 1 -- Blank

Column 2 -- X, the level number.

X = 0,1,...,A,...,Z

X = blank implies last-named level is to apply. Levels must be named in consecutive descending order.

Column 3 -- Y, section number.

Y = 5,4,3,2,1

Y = blank implies last-named section

Column 4 -- Switch up setting. SV=1

SU = 1 implies process all files in level before going to next higher level.

SU = 0 implies process one file in level X, then go to level X+1.

Column 5 -- Switch down setting. SV=1

SD = 1 or 0

Switch settings should be named just once per level.

Column 6 -- Blank

Column 7 -- Begin black box name

The remainder of the card is free-field. There are at most two fields, each with sub-fields. Fields are separated by one or more blank columns; sub-fields, by a comma. A field must be complete on one card.

#### Field 1

Sub-field 1 (must be present and begin in column 7) --  
black box name and use number

Sub-field 2 (optional)--a string of letters designating the output time-function channels, from the current box, which are to be held in jungle unit storage in floating point. Since the maximum number of output channels is 18, only letters A through R are legal.

Field 2 (Optional, according to the black box input requirements)

Each sub-field names one input channel (field or time-function) to the current box, plus its feedback requirement, if any.

Field names must precede time-function names. No channel designation implies channel A. Feedback requirements are specified within parentheses.

Refer to the appropriate black box report.

An example topology card is:

Column 1 2 3 4 5 6 7,...

0 3 EXAMP0,AC TEST0,DEL1A,LED4B(5)

Box EXAMP, use 0, is in level 0, section 3. Output channels (time-functions) A and C are to be held in floating point. Input channels are to receive data from box TEST, use 0, channel A, from box DEL, use 1, channel A, and from LED, use 4, channel B. The third input channel has a feedback offset of 5; other input channels have no feedback.

#### Data Card Format

Data cards are read by CARDRD. Data formats as allowed by FORTRAN are permissible here -- integers (decimal or octal, octal being specified by a leading zero), real constants (floating point numbers, which must include a decimal point) single or double precision (double precision being designated by a "D" immediately following the number, preceding the exponent), logical values, and field data. Refer to FORTRAN IV manual, pp 3.1 ff.

Cards are completely free-field, with either a "=" or "," separating one field from the next. Spaces and parentheses are ignored; they may be used at will for clarity.

The first field of a data card must designate the black box, including use number, into whose parameter packet the following data on the card are to be read. All other fields contain the input data, ordered as specified by the box. The first is read into the first slot in the appropriate parameter packet (or two slots, if the data is double precision), the second into the next, and so on.

Scanning of a card for data is terminated by a ";" in any column, including the first, or following column 80. Comments as desired may follow a semi-colon.

Two characters, P and R, have special functions on the data cards. Each of these, followed by an integer, n (octal or decimal), occupies a field and controls the data reads as follows: "Rn" causes the field last read and stored to be stored in the following n locations (or 2n locations if double precision). "Pn" causes the storage "pointer" to advance to the n-th slot in the packet (the first packet location is numbered 1) and begin storage of the following data in that location.

A sample input parameter card is:

EXAMPO=17,(1.6384E4,R1),16384.0DO,2'HEAD 7 890',.S.;DATA CARD

In the parameter packet for box EXAMP, use 0, will be stored:

WORD 1	000000000021
2	217400000000
3	217400000000
4	201740000000
5	000000000000
6	151206110567
7	057071600505
8	000000000001



6500 TRACOR LANE, AUSTIN, TEXAS 78721

### Error Diagnostics

There are approximately 50 different error messages available in the SONARCON programs. Most are self-explanatory. (A complete list of the diagnostics will be included later in the more extensive SONARCON report.)

The detection of an error results in a diagnostic print-out and abortion of the run.

All input cards, topology and data, are checked as completely as possible. Should an error be detected, the error is noted by the appropriate diagnostic message and all remaining input cards, including all data sets for the run, are read and syntax checked before the run is terminated.

Below are the current error messages that may be seen.

#### SØNAR1 Error Messages:

1. ERRØR, LEVEL CØDE IS ILLEGAL
2. ERRØR, SECTIØN CØDE IS ILLEGAL
3. ERRØR, LEVEL/SECTIØN MUST ØCCUR IN DECREASING ØRDER
4. ERRØR, BØX NAME/USE NUMBER EXCEEDS 7 CHARACTERS
5. ERRØR, BØX NAME FIELD NEVER TERMINATED
6. ERRØR, CHARACTER IN FLØATING PØINT CHANNEL DESIGNATØR IS ILLEGAL
7. ERRØR, CHANNEL NAME/USE NUMBER/CHANNEL DESIGNATØR EXCEEDS 8 CHARACTERS
8. ERRØR, A DIGIT ØF THE FEEDBACK SPECIFICATION IS NOT NUMERIC
9. ERRØR, VØID CHANNEL SPECIFIED
10. ERRØR, ILLEGAL CHANNEL SPECIFIED
11. ERRØR, TØPØLOGY LIST IS TØØ LARGE
12. ERRØR, CØNTRØL CARD ENCØUNTERED IN TØPØLOGY LIST
13. ERRØR, XXXXXXXX APPEARS TWICE IN THE TØPØLOGY LIST
14. ERRØR, XXXXXXXX IS LISTED AS AN INPUT CHANNEL NAME BUT IS NOT LISTED AS A BØX
15. ERRØR, A BLACK BØX NAME/USE NUMBER IS ENTIRELY NUMERIC
16. TABLE ØF CØNTENTS CANNOT BE READ FRØM DRUM

## CARDRD Error Messages:

1. ILLEGAL CONTROL CARD ENCOUNTERED
2. RUN INHIBITED
3. TOO MANY PARAMETERS FOR BLACK BOX
4. BLACK BOX NAME NOT FIRST
5. ERROR IN FIELD \_\_\_\_\_

## SONAR3 Error Messages:

1. ERROR, THE PARAMETER PACKET STORAGE REQUIREMENT IS TOO LARGE
2. THE NUMBER OF INPUT CHANNELS SPECIFIED TO XXXXXXXXXXXX, DOES NOT MATCH THE NUMBER REQUIRED
3. ERROR, THE TOPOLOGY TABLE STORAGE REQUIREMENT IS TOO LARGE

## SONARX Error Messages:

1. ERROR1
2. ERROR. INPUT FIELD TO \_\_\_\_\_ NOT AVAILABLE
3. ERROR. INSUFFICIENT FIELD STORAGE AVAILABLE FOR RUN
4. ERROR. INPUT TIME FUNCTION TO \_\_\_\_\_ NOT AVAILABLE
5. ERROR4
6. ERROR. ILLEGAL FEEDBACK CHANNEL NAMED
7. ERROR IN SPECIFICATION OF A AND/OR B-- BOX
8. ERROR1 IN COUNTF/DELETE S-R
9. ERROR2 IN COUNTF/DELETE S-R
10. ERROR. INSUFFICIENT J.UNIT STORAGE AVAILABLE FOR RUN
11. ERROR. INPUT T-F CHANNELS TO \_\_\_\_\_ ARE NOT OF EQUAL LENGTH
12. ERROR. JM FOR \_\_\_\_\_ TOO BIG
13. ERROR. C.S. TOO LONG FOR NORMAL ENTRY TO \_\_\_\_\_.
14. ERROR. T-F STG ALLOCATED IS TOO LITTLE FOR \_\_\_\_\_.
15. ERROR8 IN SECNN S-R
16. ERROR. TOO MANY CHANNELS WITH FEEDBACK SPECIFIED
17. ERROR. ILLEGAL NO. OUTPUT SAMPLES SPECIFIED BY \_\_\_\_\_.

*Howard B.S.*  
MOST Project



6500 TRACOR LANE, AUSTIN, TEXAS 78721

Addendum to

1108 SONARCON REPORT

FIELD OVERLAY CAPABILITY

25 April 1967

Date of

Document No. 67-137-U

January 18, 1967

Programmers:  
H. R. Courts  
J. G. Fairey  
J. C. Humphrey

*DR-12*  
**DISTRIBUTION STATEMENT A**

Approved for public release;  
Distribution Unlimited



6500 TRACOR LANE, AUSTIN, TEXAS 78721

## FIELD OVERLAY CAPABILITY

The SONARCON system has been extended to allow the user to specify the overlaying of field storage in core during execution of a run, thus allowing for the considerably greater total storage requirements of some runs. Care must be exercised by the user that one field is overlaid only by another which is generated following the last reference to the first field.

Field overlay specifications are named following the topology list and the 'XQT SONAR2' control card, and prior to the first input data set. The same set of overlay specifications applies to all data sets for a run.

Field overlays are specified in blocks; the full block must fit in consecutive available memory, either high or low core, or the overlay block is ignored and the run attempted without that overlay.

Field overlay blocks are assigned storage immediately following the working time function storage area. (See SONARCON Report, pages 8 ff.) SONARX attempts first to put the overlay blocks in high core. Fields are assigned in the order named in the overlay list. Note that it is thus possible to use this feature to force often-used field data into the data bank (high core), thereby minimizing run time.

### Field Overlay Deck Format

The entire overlay deck is optional. If no field overlays are desired, the input data deck immediately follows the 'XQT SONAR2' control card. When it is present the overlay



6500 TRACOR LANE, AUSTIN, TEXAS 78721

deck is read by SONAR3, which builds a temporary table. This table is later read by SONARX, which uses the information in allocating storage.

The format for each overlay card is as follows:  
Column 1 must contain an asterisk, \*.  
Columns 2 - 80 are free field. All spaces are ignored. Reading of a card is terminated by a ";" or an "\*" (see below).

Each field is the name of a black box output field, including the use number and channel name; if no channel is specified, it is interpreted as 'A'. A single field should not be named more than once.

Fields are separated by a minus sign or by one or more commas. The "-" causes the two fields to be assigned to consecutive storage. Each "," causes the storage assignment counter to back up one field in the overlay list and overlay that field's storage with the next field named.

Naming a block of fields for interdependent overlays may require more than one card. This is accomplished by terminating all but the last card of the set with an "\*". Any commas meant to follow the last-named field on a card must be on that same card, preceding the final asterisk; the continuation card must have an "\*" in column one and begin thereafter with a field name. No connecting symbol preceding the end asterisk is interpreted as a "-".

The overlay deck, when present, must be terminated by an EOF card.

#### Error Diagnostics

One or more of the following error messages will result from errors in the overlay input deck:

'ILLEGAL CARD IN FIELD OVERLAY DECK. CARD IGNORED'  
'CONTINUE FLAG ON LAST CARD IGNORED'



6500 TRACOR LANE, AUSTIN, TEXAS 78721

'TOO MANY COMMAS. CARD IGNORED'  
'ILLEGAL FIELD NAME. CARD IGNORED'  
'FIELD NAMED MORE THAN ONCE FOR OVERLAY. CARD IGNORED'  
'ERROR. FIELD OVERLAY BLOCK TOO LARGE--CARD IGNORED.  
FIELD FROM BOX XXXXX'

Example:

The following cards will cause field storage allocation as diagrammed:

\* FLDXOE,FLDXOF,FLDOG,FLDXOH,FLDXOI,FLDXOJ  
\* FLDXO-FLDXOB-FLDXOC-FLDXOD,\*,\*  
\* FLDX1A-FLDX1B  
\* SET0-SET1  
7 EOF  
8

